[예제 12-1] server: ex12-01s.c

```
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>

#define SIZE    sizeof(struct sockaddr_in)

main()
{
    int sockfd_listen;
    int sockfd_connect;

    char c;
    struct sockaddr_in server = {AF_INET, 5000, INADDR_ANY};

    printf("socket()\n");
    sockfd_listen = socket(AF_INET, SOCK_STREAM, 0);

    printf("bind()\n");
    bind(sockfd_listen, (struct sockaddr *)&server, SIZE);

    printf("listen()\n");
    listen(sockfd_listen, 5);

    printf("wating for client\n");
    sockfd_connect = accept(sockfd_listen, NULL, NULL);
    printf("accepted\n");

    recv(sockfd_connect, &c, 1, 0);
    printf("recv %c from client\n", c);

    c++;
    printf("send %c to client\n", c);
    send(sockfd_connect, &c, 1, 0);

    printf("close()\n");
    close(sockfd_connect);
    close(sockfd_listen);
}
```

[예제 12-1-1] client: ex12-01c.c

```
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>

#define SIZE    sizeof(struct sockaddr_in)

main()
{
    int sockfd;
    char send_c, recv_c;
    struct sockaddr_in server = {AF_INET, 5000};

    server.sin_addr.s_addr = inet_addr("127.0.0.1");

    printf("socket()\n");
    sockfd = socket(AF_INET, SOCK_STREAM, 0);

    printf("connect()\n");
    connect(sockfd, (struct sockaddr *)&server, SIZE);

    send_c = 'A';

    printf("send %c to server\n", send_c);
    send(sockfd, &send_c, 1, 0);

    recv(sockfd, &recv_c, 1, 0);
    printf("recv %c from server\n", recv_c);

    printf("close()\n");
    close(sockfd);
}
```

[예제 12-2] ex12-02.c

```
#include <arpa/inet.h>
#include <unistd.h>

main()
{
    char *valid = "197.0.0.1";
    char *invalid = "300.0.0.1";

    in_addr_t ipaddr1;
    struct in_addr ipaddr2;

    if((ipaddr1 = inet_addr(valid)) == -1)
        printf("invalid: %s\n", valid);
    else
        printf("valid: %d.%d.%d.%d\n",
           (ipaddr1 >>  0) & 0xFF,
           (ipaddr1 >>  8) & 0xFF,
           (ipaddr1 >> 16) & 0xFF,
           (ipaddr1 >> 24) & 0xFF );

    ipaddr2.s_addr = ipaddr1;

    if((ipaddr1 = inet_addr(invalid)) == -1)
        printf("invalid: %s\n", invalid);
    else
        printf("vaild: %x\n", ipaddr1);

    printf("%s\n", inet_ntoa(ipaddr2));
}
```

[예제 12-3] ex12-03.c

```
#include <arpa/inet.h>
#include <sys/socket.h>
#include <unistd.h>

main()
{
        int sockfd;

        if( (sockfd = socket(AF_INET, SOCK_STREAM, 0)) == -1)
        {
                printf("fail to call socket()\n");
                exit(1);
        }

        printf("socket descriptor is %d\n", sockfd);

        /* 소켓을 통한 통신 기능을 수행한다. */

        close(sockfd);
}
```

[예제 12-4] ex12-04.c

```
#include <sys/socket.h>
#include <unistd.h>
#include <netinet/in.h>

#define SIZE    sizeof(struct sockaddr_in)

main()
{
    int sockfd;
    struct sockaddr_in addr;

    addr.sin_family = AF_INET;
    addr.sin_port = 1004;
    addr.sin_addr.s_addr = INADDR_ANY;

    /* socket()으로 소켓을 생성하는 코드 */

    if(bind(sockfd, (struct sockaddr *)&addr, SIZE) == -1)
    {
            printf("fail to call bind()\n");
            exit(1);
    }

    /* 통신을 위한 나머지 코드들.. */

    close(sockfd);
}
```

[예제 12-5] ex12-05.c

```
#include <...h>

#define SIZE    sizeof(struct sockaddr_in)

main()
{
    int sockfd_listen;
    char c;
    struct sockaddr_in server = {AF_INET, 5000, INADDR_ANY};

    /* socket()을 호출하는 부분 */

    /* bind()를 호출하는 부분 */

    if(listen(sockfd_listen, 5) == -1) {
        printf("fail to call listen()\n");
        exit(1);
    }

    /* 통신을 수행하는 부분 */
}
```

[예제 12-6] ex12-06.c

```
#include <….h>

#define SIZE    sizeof(struct sockaddr_in)


int sockfd_connect;

main()
{
    int sockfd_listen;
    char c;
    struct sockaddr_in server = {AF_INET, 5000, INADDR_ANY};

    if((sockfd_listen = socket(AF_INET, SOCK_STREAM, 0)) == -1) {
        printf("fail to call socket()\n");
        exit(1);
    }

    if(bind(sockfd_listen, (struct sockaddr *)&server, SIZE) == -1) {
        printf("fail to call bind()\n");
        exit(1);
    }

    if(listen(sockfd_listen, 5) == -1) {
        printf("fail to call listen()\n");
        exit(1);
    }

    while(1) {
        if((sockfd_connect = accept(sockfd_listen, NULL, NULL)) == -1) {
            printf("fail to call accept()\n");
            continue;
        }

        /* sockfd_connect를 사용하여 통신을 수행 */

    }
}
```

[예제 12-7] ex12-07.c

```
#include <….h>

#define SIZE    sizeof(struct sockaddr_in)

main()
{
    int sockfd;
    char send_c, recv_c;
    struct sockaddr_in server = {AF_INET, 5000};

    server.sin_addr.s_addr = inet_addr("127.0.0.1");

    if((sockfd = socket(AF_INET, SOCK_STREAM, 0)) == -1) {
        printf("fail to call socket()\n");
        exit(1);
    }

    if(connect(sockfd, (struct sockaddr *)&server, SIZE) == -1) {
        printf("fail to call connect()\n");
        exit(1);
    }

    /* 메시지를 주고받는 부분 */
}
```

[예제 12-8] ex12-08.c

```
#define SIZE    sizeof(struct sockaddr_in)

int sockfd_connect;

main()
{
    /* sock_listen 소켓 생성 */
    /* sock_listen으로 bind 호출 */
    /* sock_listen으로 listen 호출 */

    while(1) {

        /* accept를 호출하여 sockfd_connet를 생성 */

        while(recv(sockfd_connect, &c, 1, 0) > 0)
            send(sockfd_connect, &c, 1, 0);

        close(sockfd_connect);
    }
}
```

[예제 12-9] ex12-09.c

```
...

while(1) {
    sockfd_connect = accept(sockfd_listen, NULL, NULL);

    /* send, recv를 호출하는 부분 */

    close(sockfd_connect);
}

...
```

[예제 12-10] server: ex12-10s.c

```
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <signal.h>

#define SIZE    sizeof(struct sockaddr_in)

void closesock(int sig);

int sockfd_connect;

main()
{
    int sockfd_listen;
    char c;
    struct sockaddr_in server = {AF_INET, 5000, INADDR_ANY};
    struct sigaction act;

    act.sa_handler = closesock;
    sigfillset(&(act.sa_mask));
    sigaction(SIGPIPE, &act, NULL);

    if((sockfd_listen = socket(AF_INET, SOCK_STREAM, 0)) == -1) {
        printf("fail to call socket()\n");
        exit(1);
    }

    if(bind(sockfd_listen, (struct sockaddr *)&server, SIZE) == -1) {
        printf("fail to call bind()\n");
        exit(1);
    }

    if(listen(sockfd_listen, 5) == -1) {
        printf("fail to call listen()\n");
        exit(1);
    }

    while(1) {
```

```
        if((sockfd_connect = accept(sockfd_listen, NULL, NULL)) == -1) {
            printf("fail to call accept()\n");
            continue;
        }
        printf("accepted\n");
while(recv(sockfd_connect, &c, 1, 0) > 0)
            send(sockfd_connect, &c, 1, 0);

        printf("close(sockfd_connect)\n");
        close(sockfd_connect);
    }
}

void closesock(int sig)
{
    close(sockfd_connect);
    printf("connection is lost\n");
    exit(0);
}
```

[예제 12-10-1] client: ex12-10c.c

```
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>

#define SIZE    sizeof(struct sockaddr_in)

main()
{
    int sockfd;
    char send_c, recv_c;
    struct sockaddr_in server = {AF_INET, 5000};

    server.sin_addr.s_addr = inet_addr("127.0.0.1");

    if((sockfd = socket(AF_INET, SOCK_STREAM, 0)) == -1) {
        printf("fail to call socket()\n");
        exit(1);
    }

    if(connect(sockfd, (struct sockaddr *)&server, SIZE) == -1) {
        printf("fail to call connect()\n");
        exit(1);
    }

    recv_c = '\n';
    while(1) {
        if(recv_c == '\n')
           printf("Input a message\n");
        send_c = getchar();

        send(sockfd, &send_c, 1, 0);

        if(recv(sockfd, &recv_c, 1, 0) > 0)
            printf("%c", recv_c);
        else
        {
            printf("server has no reply\n");
            close(sockfd);
            exit(1);
```

```
            }
        }
    }
```